**API Documentation**

| **Function** | **Function Description** | **Inputs / Outputs** |
| --- | --- | --- |
| def read\_file(filename): | This function is responsible for reading the contents of the user specified file | **Input**: filename (str) - Name of the input file specified by the user.  **Output:** chars (str) - Content of the file. |
| def write\_file(output\_Filename, output\_data): | This function is responsible for writing the results of this program to a new file, the name of the file will be specified by the user. | **Input:** output\_Filename (str) - Name of the output file specified by the user.  output\_data (str) - Results of the program to be written to the output file.  **Output:** None / returns the results of the program in the output file |
| def wordCount(chars): | This function counts the number of total words in the user-specified file. | **Input:** chars (str) - Content of the input file.  **Output:** wordCount (int) - The total count of words in the file. |
| def num\_sentences(chars): | This function counts the total number of sentences in the file, considering periods and question marks as sentence delimiters. | **Input:** chars (str) - Content of the input file.  **Output:** num\_of\_sent (int) - The total number of sentences in the input file. |
| def frequency(chars): | This function counts the total number of times each word appears in the input text file, considering word frequencies and removing stopwords and punctuation. | **Input:** chars (str) - Content of the input file.  **Output:** word\_freq (dict) - The total number of times each word appears in the input file. |
| def user\_word\_frequency(chars): | This function counts the total number of times a specific word, which is specified by the user, appears in the file. | **Input:** chars (str) - Content of the input file.  **Output:** freq\_of\_word (str) - The frequency of the word specified by the user. |
| def frequent\_words(chars): | This function counts the total number of times a word appears in the file, sorts them in descending order, and outputs the top 10 most occurring words. | **Input:** chars (str) - Content of the input file.  **Output:** df (DataFrame) - A DataFrame consisting of the top 10 words that occurred most frequently in the file. |
| def termFrequency\_idf(chars): | This function calculates the TF-IDF value for each word in the document and displays it as a DataFrame. | **Input:** chars (str) - Content of the input file.  **Output**: tfidf\_df (DataFrame) - A DataFrame containing the Term Frequency-Inverse Document Frequency (TF-IDF) of each word in the document. Also, a new file called **TermFrequcny.csv** will be created and it will contain the  Term Frequency - Inverse Document Frequency (TF-IDF) of each word in the document. |
| def avg\_words\_in\_sent(chars): | This function calculates the average number of words in a sentence in the input file, returning 0 if the file is empty. | **Input:** chars (str) - Content of the input file.  **Output**: avg\_word\_count (float) - The average number of words in a sentence in the entire file. |
| def longest\_word(chars): | This function returns the longest word in the text file. | **Input**: chars (str) - Content of the input file.  **Output:** longest (str) - The longest word in the input file. |
| def length\_longest\_word(chars): | This function returns the length of the longest word in the text file. | **Input:** chars (str) - Content of the input file.  **Output**: len(longest) (int) - The length of the longest word in the input file. |
| def shortest\_word(chars): | This function returns the shortest word in the text file. | **Input:** chars (str) - Content of the input file.  **Output:** shortest\_word (str) - The shortest word in the input file. |
| def count\_chars(chars): | This function calculates the frequency of each character in the file and saves it in a DataFrame. | **Input:** chars (str) - Content of the input file.  **Output:** table (Pandas DataFrame) - A DataFrame containing the frequency of each character in the file and saves it in a DataFrame. Also, a new file called **alphabet\_frequcny.csv** will be created and it will contain the frequency of each "character" in the file. |